

VŠB – Technická univerzita Ostrava
Univerzitní studijní programy

Aplikace Raspberry Pi 2 s dotykovým LCD displejem
Application of Raspberry Pi 2 with LCD touchscreen

Diplomant:

Vladan Najdek

Vedoucí bakalářské práce:

doc. Ing. Marek Babiuch, Ph.D.

Rok odevzdání:

2016

Zadání bakalářské práce

Student: **Vladan Najdek**
Studijní program: **B3943 Mechatronika**
Studijní obor: **3906R006 Mechatronické systémy**
Téma: **Aplikace Raspberry Pi 2 s dotykovým LCD displejem**
Application of Raspberry PI2 with LCD Touchscreen
Jazyk vypracování: **čeština**

Zásady pro vypracování:

1. Charakterizujte mikropočítač Raspberry Pi2, jeho vývojové verze, hardwarové parametry a možnosti softwarové podpory.
2. Dokumentujte postup zprovoznění operačních systémů Raspbian Linuxu a Windows 10 IoT na zařízení Raspberry Pi2 (včetně přípravy image obrazů na microSD karty a následné instalace).
3. Popište dostupné periferie pro Raspberry Pi2, možnosti Wi-fi připojení a oficiální dotykový displej a jejich zprovoznění.
4. Popište současné vývojářské možnosti se systémem Raspberry Pi2, možnosti vývoje aplikací v různých technologiích a jazycích (Python, C#, ASP.NET a dalších)
5. Navrhněte a implementujte demonstrační aplikaci využití dotykového displeje pro Raspberry Pi2.
6. Zhodnoťte dosažené výsledky a navrhněte možné využití dotykového displeje v dalších typech aplikací.

Seznam doporučené odborné literatury:

Microsoft Dev Center.Windows IoT – Setup your Raspberry Pi 2. Dostupné z:
<http://ms-iot.github.io/content/en-US/win10/SetupPCRPI.htm>
Microsoft Dev Center. Downloads and Tools. Dostupné z:
<https://ms-iot.github.io/content/en-US/Downloads.htm>
Kivy Pie Kivy onRaspberry PI - Python framework for multitouch apps. Dostupné z:
<http://kivypie.mitako.eu/kivy-pie.html>
Nixon, D. Getting Started with Python and Raspberry Pi. Pack Publishing, Open Source Book- Community Experience Distilled, 2015, 303 stran.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marek Babiuch, Ph.D.**

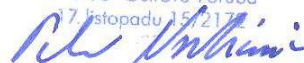
Datum zadání: 20.10.2015

Datum odevzdání: 16.05.2016



doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry

VYSOKÁ ŠKOLA BAŇSKÁ
TECHNICKÁ UNIVERZITA OSTRAVA
708 33 Ostrava-Poruba
17. listopadu 15/2172



prof. Ing. Petr Noskievič, CSc.
prorektor pro studium

Prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě:

.....

podpis studenta

Prohlašuji, že:

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že bakalářská práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě dne:

.....

podpis studenta

Jméno a příjmení autora práce:

Vladan Najdek

Adresa trvalého pobytu autora práce:

Matušínského 1044/8 716 00 Ostrava – Radvanice

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce doc. Ing. Marku Babiuchovi, Ph.D. za pomoc a podporu při vypracování této práce.

Anotace

NAJDEK, V. *Aplikace Raspberry Pi 2 s dotykovým LCD displejem: bakalářská práce*. Ostrava: VŠB – Technická univerzita Ostrava, Univerzitní studijní programy, 2016, 47 s. Vedoucí práce: doc. Ing. Marek Babiuch, Ph.D.

Tato práce popisuje možnosti využití Raspberry Pi 2 v kombinaci s oficiálním dotykovým displejem pro Raspberry Pi. Nejprve jsou detailně popsány hardwarové vlastnosti Raspberry Pi 2 včetně srovnání s předchozími verzemi. Následuje rozbor operačních systémů, které lze při práci s tímto zařízením využívat, především Raspbianu a Windows IoT, a popis jejich instalace. Dále je dokumentováno několik periférií a jejich zprovoznění, se zaměřením na oficiální dotykový displej a možnosti připojení k internetu. Nakonec jsou popsány možnosti vývoje aplikací v různých technologiích a jazycích, především Kivy a Mono. V Kivy je také vytvořena aplikace, která demonstruje možnosti využití Raspberry Pi 2 s dotykovým displejem.

Klíčová slova

Raspberry Pi 2, dotykový displej, Kivy, Raspbian, Python

Annotation

NAJDEK, V. *Application of Raspberry Pi 2 with LCD touchscreen: bachelor thesis*. Ostrava: VŠB – Technical University of Ostrava, University Study Programmes, 2016, 47 p. Thesis head: doc. Ing. Marek Babiuch, Ph.D.

This thesis describes the possibilities of using Raspberry Pi 2 in combination with the official touchscreen for Raspberry Pi. In the beginning it deals with operating systems which can be used with this device especially Raspbian and Windows IoT and their installation process. Then there is a documentation of different kinds of peripheral devices and their commissioning with main focus on the official touchscreen and the internet connection. In the end there is a description of possibilities of application development in different technologies and programming languages, especially Kivy and Mono. In Kivy I have created an application that demonstrates the possibilities of Raspberry Pi 2 with a touchscreen.

Keywords

Raspberry Pi 2, touchscreen, Kivy, Raspbian, Python

Obsah

Seznam použitých zkratk	8
Seznam použitých značek	9
Úvod	10
1 Základní charakteristika	11
1.1 Vývojové verze	12
1.2 Softwarová podpora zařízení	13
2 Postup zprovoznění vybraných operačních systémů	15
2.1 Windows 10 IoT	15
2.2 Raspbian	18
3 Periferie	20
3.1 Oficiální Raspberry Pi displej	20
3.2 Připojení k WiFi	23
4 Vývojářské možnosti	24
4.1 Kivy framework	25
4.2 Mono framework	27
5 Návrh modelu pro demonstrační aplikaci	30
5.1 Obsluha dveří a vrat	30
5.2 Obsluha osvětlení	32
5.3 Obsluha ventilátoru	33
6 Návrh grafického rozhraní demonstrační aplikace	35
6.1 Úvodní obrazovka	35
6.2 Hlavní obrazovka	36
7 Návrh programové logiky demonstrační aplikace	40
Závěr	44
Literatura	46
Seznam příloh	48

Seznam použitých zkratek

3D	– Trojrozměrné (Three-dimensional)
CSI	– Kamerové sériové rozhraní (Camera serial interface)
DSI	– Displejové sériové rozhraní (Display serial interface)
DVI	– Digitální vizuální rozhraní (Digital visual interface)
GND	– Uzemnění (Ground)
GPIO	– Obecný vstup/výstup (General-purpose input/output)
GPU	– Grafický procesor (Graphics processing unit)
GUI	– Grafické uživatelské rozhraní (Graphical user interface)
HDMI	– Multimediální rozhraní s vysokým rozlišením (High-definition multi-media interface)
I ² C	– Typ sériové sběrnice (Inter-integrated circuit)
IDE, IDLE	– Integrované vývojové prostředí (Integrated development environment)
IoT	– Internet věcí (Internet of things)
LCD	– Displej z tekutých krystalů (Liquid crystal display)
LED	– Světlo emitující dioda (Light emitting diode)
NOOBS	– Manažer instalace operačního systému (New out of box software)
PS/2	– Typ konektoru periferií
PWM	– Pulzně šířková modulace (Pulse width modulation)
PWR	– Napájení (Power)
RAM	– Paměť s přímým přístupem (Random-access memory)
SoC	– Systém na čipu (System-on-a-chip)
SPI	– Sériové periferní rozhraní (Serial peripheral interface)
SSH	– Zabezpečený komunikační protokol (Secure shell)
SSID	– Identifikátor bezdrátové sítě (Service set identifier)
USB	– Univerzální sériová sběrnice (Universal serial bus)

Seznam použitých značek

° – Stupeň – jednotka velikosti úhlu

A – Ampér – jednotka proudu

Hz – Hertz – jednotka frekvence

m – metr – jednotka délky

mA – miliampér – jednotka proudu (10^{-3} A)

MB – Megabyte – jednotka množství dat

Mb/s – Megabit za sekundu – jednotka rychlosti přenosu dat

MHz – Megahertz – jednotka frekvence (10^6 Hz)

mm – milimetr – jednotka délky (10^{-3} m)

ms – milisekunda – jednotka času (10^{-3} s)

s – sekunda – jednotka času

Úvod

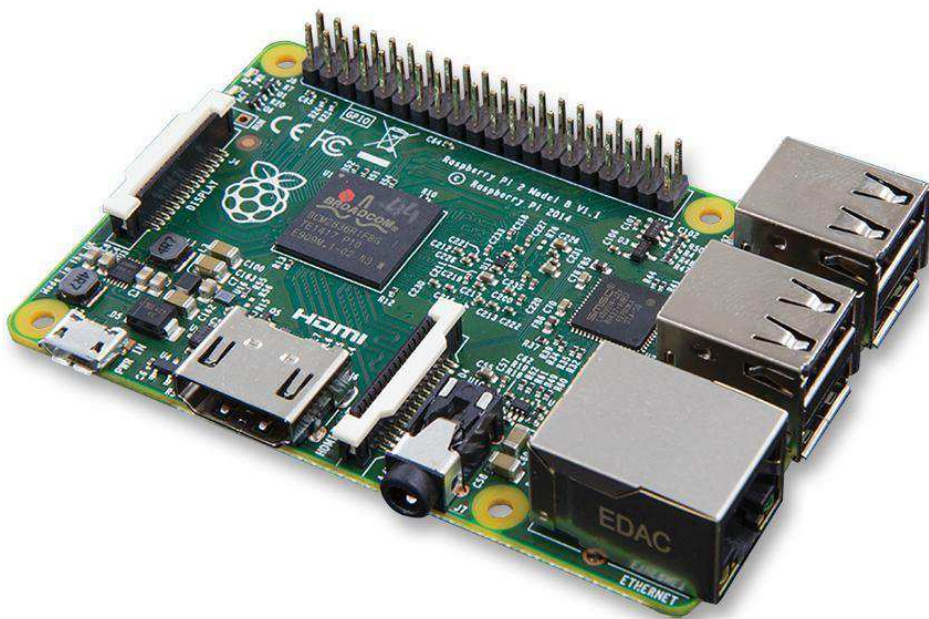
Raspberry Pi 2 je model mikropočítače od Raspberry Foundation. Cílem této nadace bylo vytvořit malý a cenově dostupný mikropočítač za účelem vzdělávání v oblasti programování. Nicméně jeho nízká cena a velikost, při zachování obstojného výkonu, mu umožňuje do určité míry konkurovat klasickým stolním počítačům. Raspberry Pi 2 rozhodně není schopno provádět graficky nebo výpočetně náročné operace jako třeba 3D modelování, zato je ale ideální pro jednoduchá použití typu multimediální centrum pro přehrávání hudby a filmů nebo psaní dokumentů a e-mailů.

V porovnání se stolním počítačem nabízí Raspberry Pi navíc GPIO piny, ke kterým lze připojit elektromotorky, LED diody a jiné součástky. To umožnilo vytvoření mnoha různých druhů robotů a jiných elektrotechnických projektů. Z pomůcky pro výuku programování se tedy zároveň stala pomůcka pro vzdělávání v oblasti elektrotechniky. Všechna tato využití stojí za tím, že se z Raspberry Pi stal světový fenomén, kterého se prodalo přes pět milionů kusů.

V této práci bych chtěl toto zařízení podrobně prozkoumat a rozebrat možnosti využití, které nabízí v kombinaci s dotykovým LCD displejem. Pro demonstraci vytvořím jednoduchou aplikaci, která bude ovládat model automatizované domácnosti.

1 Základní charakteristika

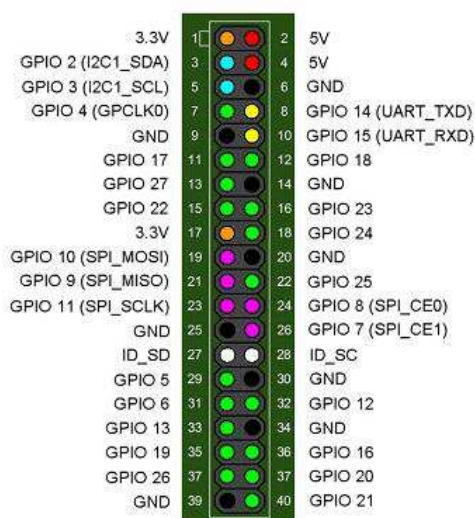
Raspberry Pi 2 je jeden z nejmenších, běžně dostupných počítačů. Jeho rozměry jsou pouhých 85x56x21 mm. Vzhledem k velikosti by nebylo vhodné použít klasickou základní desku osazenou komponenty, proto je využita technologie SoC, která sdružuje všechny podstatné komponenty, jako CPU, GPU nebo ovladače USB, do jediného čipu. Toto zařízení používá SoC BCM2836, který obsahuje čtyřjádrový procesor ARM Cortex-A7 complex s frekvencí 900 MHz a grafický procesor Videocore 4. Tato grafická jednotka zvládne přehrávat video v Blu-ray kvalitě ve formátu H.264 při rychlosti 40 Mb/s a obsahuje rychlé 3D jádro, dostupné pomocí dodaných OpenGL ES 2.0 a OpenVG knihoven. Na desce se rovněž nachází RAM o velikosti 1024 MB. [Raspberry Pi, 2015a]



Obr. 1.1: Raspberry Pi 2 [Exp-tech, 2015]

Pro připojení klávesnice a myši, případně dalších periferií nabízí tento model 4 USB 2.0 porty. Připojení k internetu lze realizovat pomocí 10/100 Ethernet portu nebo připojením WiFi adaptéru k USB portu. Pro komunikaci s monitorem či televizí je dostupný HDMI výstup. K přenosu zvuku slouží kombinovaný 3,5mm jack. Ten také umožňuje přenos kompozitního videa např. pro starší televize. Některé LCD displeje mohou být také připojeny k DSI konektoru. Na desce se rovněž nachází CSI port, který vypadá stejně jako DSI port a slouží pouze k připojení kamery. Pro aplikace lze využít

GPIO port se 40 piny. Z obr. 1.2 je patrné, že se zde nacházejí i specializované piny např. pro I²C sběrnici nebo SPI sběrnici a další. [Raspberry Pi, 2015a]



Obr. 1.2: Rozložení GPIO pinů [Designspark, 2015]

1.1 Vývojové verze

V současné době je stále možné koupit starší verze Raspberry Pi. Aplikace vyvinuté pro jednotlivé verze jsou plně kompatibilní, není tedy problém spustit na Raspberry Pi 2 jakoukoliv dříve vyvinutou aplikaci. [Raspberry Pi, 2015a]

Prvním modelem je Raspberry Pi B. Poprvé byl oznámen v dubnu 2012, v prodeji od června téhož roku. Využívá SoC Broadcom BCM2835 s pouze jednojádrovým procesorem ARM1176JZF-S, běžícím na frekvenci 700 MHz, a Videocore 4 GPU. Velikost RAM je 512 MB. Připojení k internetu lze realizovat pomocí 10/100 Ethernet portu nebo připojením WiFi adaptéru k jednomu ze dvou USB 2.0 portů. Obraz je možné přenášet HDMI, RCA nebo DSI výstupem. Na desce lze nalézt i CSI pro připojení kamery. Pro připojení dalších zařízení je připraven GPIO port s 24 piny. O rok později byl vydán model A, který má sloužit jako levnější alternativa. Jeho výhodou je kromě ceny také menší velikost, proto se hodí zejména pro projekty, při kterých běží zařízení samostatně. Základ je stejný jako u modelu B, ale obsahuje pouze 1 USB port, chybí Ethernet port a RAM má velikost pouze 256 MB. [SparkFun Electronics, 2015] [eLinux, 2015]

V červnu 2014 byl oznámen model B+. V porovnání s modelem B má o 2 USB 2.0 porty a 14 GPIO pinů více. Slot na SD kartu byl nahrazen slotem pro mikro SD kartu.

HDMI výstup zůstal, ale RCA port byl nahrazen kombinovaným portem s 3,5mm audio jackem. Zlepšením je rovněž menší spotřeba, která je dána nahrazením lineárních regulátorů napětí spínacími. O rok později byla vydána i vylepšená verze modelu A. Model A+ byl vylepšen podobně jako model B+. Došlo tedy ke zvýšení počtu GPIO pinů, snížení spotřeby a nahrazení slotu pro SD kartu a RCA výstupu. Výhody a nevýhody modelu A+ zůstávají stejné jako v případě modelu A, tedy 1 USB port, RAM pouze 256 MB a absence Ethernetu. [eLinux, 2015] [Designspark, 2015]

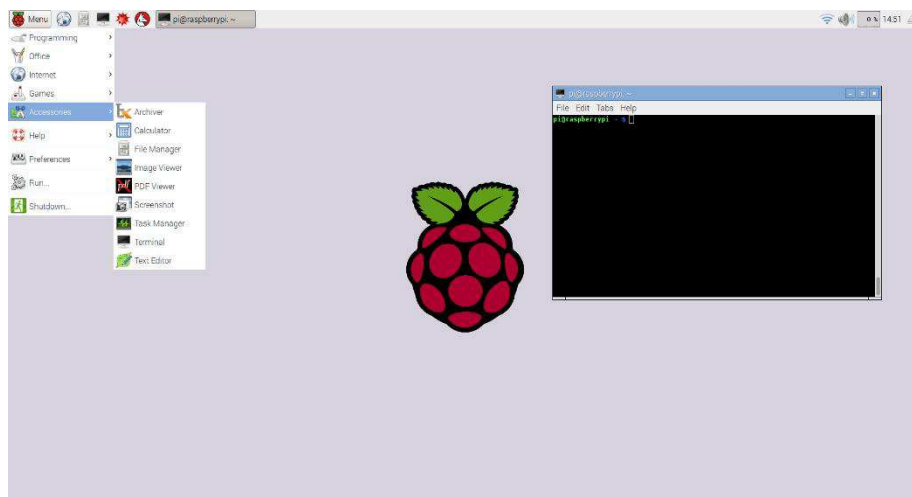
Tab. 1.1: Srovnání vývojových verzí.

	B	A	B+	A+	2 B
Frekvence CPU	700 MHz	700 MHz	700 MHz	700 MHz	4x900 MHz
Typ GPU	Videocore 4	Videocore 4	Videocore 4	Videocore 4	Videocore 4
Velikost RAM	512 MB	256 MB	512 MB	256 MB	1024 MB
Počet USB portů	2	1	2	1	4
Typ paměťové karty	SD	SD	microSD	microSD	microSD
Ethernet	10/100 Mb/s	Ne	10/100 Mb/s	Ne	10/100 Mb/s
Počet GPIO pinů	24	24	40	40	40

1.2 Softwarová podpora zařízení

Pro Raspberry Pi 2 je dostupný oficiální operační systém Raspbian, který je založený na operačním systému Debian. Lze použít i další Linuxové distribuce např. Ubuntu MATE nebo Fedora. Tyto systémy se vyznačují grafickým uživatelským rozhraním, ale lze je ovládat i pomocí příkazového řádku, který se nazývá terminál. [Element14, 2015]

Pro psaní textových dokumentů, zpracování tabulek a další kancelářské aplikace je možné využít programy softwarové sady LibreOffice, která je předinstalována v Raspbianu. Alternativně lze naistalovat i zdarma dostupnou sadu OpenOffice, nicméně Microsoft Office není podporován. LibreOffice zvládá ukládání i otevírání souborů s příponou docx.



Obr. 1.3: Uživatelské rozhraní Raspbianu [Distrowatch, 2015]

K prohlížení internetu je k dispozici webový prohlížeč Epiphany. Lze stáhnout i další prohlížeče, které mají svou linuxovou verzi jako např. Firefox. Pro přehrávání videí slouží Omxplayer. Vzhledem k tomu, že původní účel Raspberry Pi je podpora výuky, jsou v Raspbianu připravené aplikace pro programování Python IDLE a Scratch, dále pak Wolfram a Mathematica.

Novinkou pro Raspberry Pi 2 je operační systém Windows 10 IoT. Ten neobsahuje žádné aplikace a jeho po instalaci se zobrazí pouze obrazovka s IP adresou, verzí operačního systému, nastavením připojení a několika návody pro aplikace. Tento systém slouží především jako základ pro uživatele, kteří chtějí, aby běžela pouze jejich vestavěná aplikace. [Bechynský, 2015]



Obr. 1.4: Základní obrazovka Windows IoT. [Windows dev center, 2015a]

2 Postup zprovoznění vybraných operačních systémů

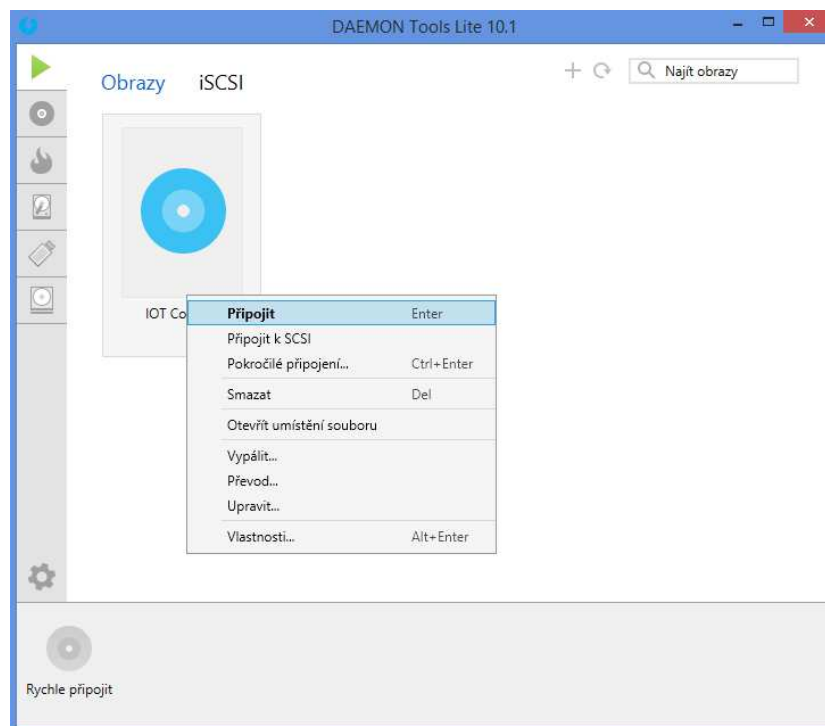
Postup instalace obou systémů se mírně liší, ale potřebné vybavení je téměř totožné. Kromě Raspberry Pi 2 a počítače, na kterém připravíme microSD kartu, budeme potřebovat monitor. Pro Raspbian je možné využít jak různé displeje pro Raspberry, tak monitor připojený k HDMI výstupu, zatímco Windows IoT zatím podporuje pouze HDMI výstup. Pokud budeme chtít připojit Raspberry k internetu, máme na výběr mezi připojením přes Ethernet a USB WiFi adaptérem. V současné době podporuje Windows IoT pouze oficiální WiFi adaptér od Raspberry. Vhod může přijít i klávesnice s myší, ačkoli pokud máme Raspberry připojené k síti, můžeme k ovládání použít SSH.

Návody uvedené níže byly ověřeny na počítači s operačním systémem Windows 8.1, platný je ovšem i pro Windows 10. Na starších verzích systému Windows by se neměly vyskytnout problémy. Pro Linuxové distribuce a OS X se postup může mírně lišit, hlavně pokud jde o použitý software. Co se týče programů, budeme potřebovat program pro zpracování ISO souborů. Od verze 7 mají operační systémy Windows pro tyto účely předinstalovanou aplikaci, ale lze použít třeba zdarma dostupný program Daemon tools lite. Dále budeme potřebovat program pro zápis ISO souboru na SD kartu. Já jsem použil při instalaci obou systémů program Win 32 Disk Imager, ačkoli pro Windows IoT je možné použít aplikaci Windows IoT Core Image Helper, která se instaluje současně s ostatními z image souboru Windows IoT.

2.1 Windows 10 IoT

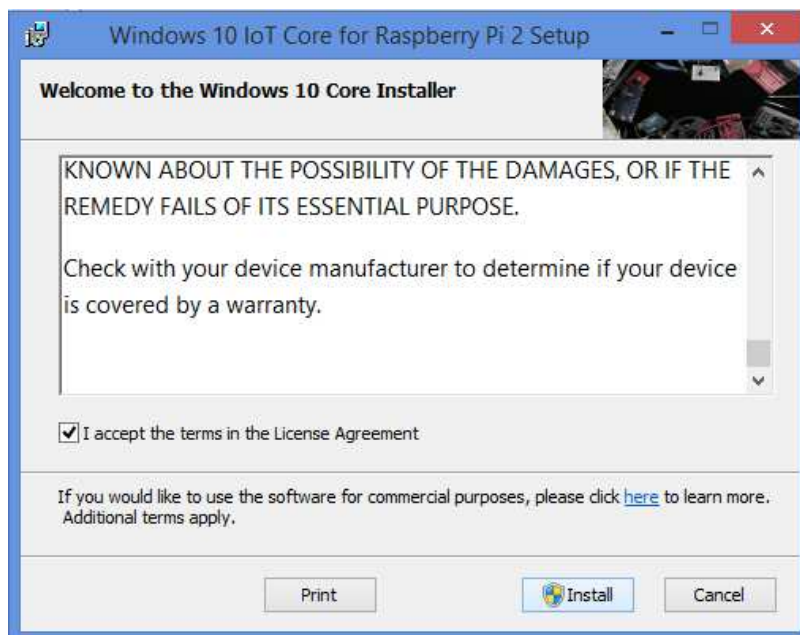
Windows 10 IoT pro Raspberry Pi 2 je dostupný od února 2015. Pro jeho instalaci je třeba provést následující kroky:

1. Nejprve si v internetovém prohlížeči otevřeme stránku
<http://ms-iot.github.io/content/en-US/Downloads.htm>
2. Klikneme na tlačítko Download RTM Release for Raspberry Pi 2. Tím stáhneme do počítače image soubor IoT Core RPi.
3. Připojíme stažený ISO soubor pomocí odpovídajícího programu.



Obr. 2.1: Prostředí programu Daemon tools.

4. Spustíme jediný instalační soubor, který se v obrazu nachází a sice Windows_10_IoT_Core_RPi2.
5. V okně, které se nám otevře, si přečteme licenční ujednání, zaškrtneme box pod ním a klikneme na tlačítko Install.



Obr. 2.2: Instalace softwaru pro přípravu SD karty.

6. Počkáme, dokud se instalace nedokončí a klikneme na tlačítko Finish.
7. Vložíme microSD kartu do čtečky a spustíme program pro zápis na kartu.

8. Vybere v seznamu kartu, na kterou chceme zapisovat a do kolonky select image vložíme C:/Program Files(x86)/MicrosoftIoT/FFU/RaspberryPi2/flash.ffu. Pomocí tlačítka Write spustíme zápis.



Obr. 2.3: Prostředí programu Windows IoT core image helper.

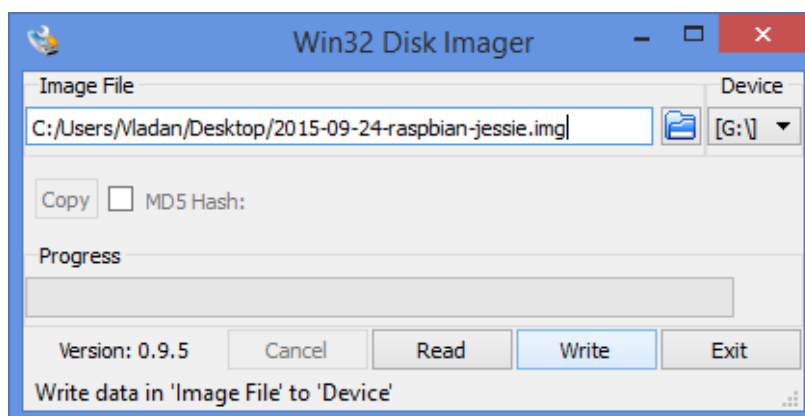
9. (Volitelný krok) Pokud nebudeme chtít mezi počítačem a Raspberry vytvořit síťové spojení programem Windows IoT Core Watcher, můžeme prostřednictvím ovládacích panelů odinstalovat program Windows 10 IoT core for Raspberry Pi 2.
10. Vložíme microSD kartu do Raspberry Pi 2 a připojíme napájení, klávesnici, myš, monitor a případně i ethernet nebo USB WiFi adaptér.
11. Windows se sám spustí. Počkáme několik minut, než se spustí přednastavená aplikace zobrazující IP adresu, verzi operačního systému a další údaje.

Windows IoT momentálně nemá předinstalované žádné aplikace. Z hlavní obrazovky můžeme pouze nastavit připojení k internetu a v záložce tutorials přehrát několik videí s příklady projektů běžících na Windows IoT. [Windows dev center, 2015b]

2.2 Raspbian

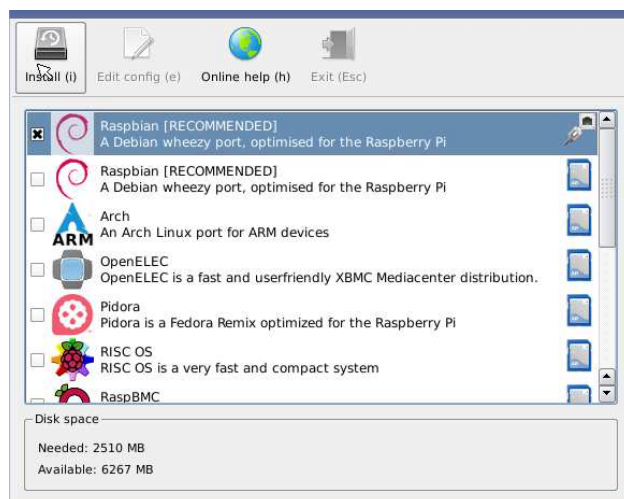
První verze Raspbianu byla vydána v květnu 2012, tedy měsíc po uvedení modelu B. V případě, že vlastníme microSD kartu s předinstalovaným softwarem NOOBS, lze začít krokem 5.

1. Nejprve si v internetovém prohlížeči otevřeme stránku
<https://www.raspberrypi.org/downloads/raspbian/>
2. Stáhneme image soubor s nejnovější verzí, kterou je momentálně Raspbian Jessie.
3. Vložíme microSD kartu do čtečky a spustíme program pro zápis na kartu.
4. Vybere v seznamu SD kartu, na kterou chceme zapisovat, do kolonky select image vložíme cestu k image souboru, který jsme stáhli a spustíme zápis.



Obr. 2.4: Prostředí programu Win32 disk imager.

5. Vložíme microSD kartu do Raspberry Pi 2 a připojíme napájení, klávesnici, myš, monitor a případně i Ethernet nebo USB WiFi adaptér.
6. Pokud jsme vložili kartu s NOOBS, zobrazí se okno s operačními systémy, které lze nainstalovat. Vybereme Raspbian a klikneme na tlačítko Install.



Obr. 2.5: Výběr operačního systému v NOOBS [Raspberry Pi, 2015c]

7. Počkáme než se Raspbian spustí a zobrazí se pracovní plocha. Nyní můžeme nastavit připojení k síti.
8. Pokud jsme instalovali pomocí NOOBS je vhodné aktualizovat systém na nejnovější verzi zadáním následujících příkazů `sudo apt-get update` a `sudo apt-get upgrade` do terminálu.

Během instalace můžeme být vyzváni k zadání uživatelského jména a hesla. Napíšeme přednastavené uživatelské jméno *pi* a heslo *raspberrypi*. Během psaní hesla se nezobrazují na obrazovce žádné znaky. To je v pořádku, jedná se o bezpečnostní opatření všech Linuxových distribucí. [Raspberry Pi, 2015b] [Raspberry Pi, 2015d]

3 Periferie

K Raspberry Pi je možné připojit mnoho různých druhů periférií. Jedná se především o periferie, na které jsme zvyklí u běžných stolních počítačů jako myši a klávesnice, ale také gamepady, joysticky nebo jiné ovládací zařízení připojitelné pomocí rozhraní USB. Některé starší klávesnice a myši mohou mít PS/2 konektor, v takovém případě je potřeba využít PS/2-USB redukce. Pokud chceme připojit zobrazovací zařízení, většina jich bude využívat HDMI výstup, ačkoliv některé jednodušší displeje mohou být připojeny pomocí GPIO portu. Výjimkou je oficiální displej pro Raspberry, který se připojuje přes DSI. Pro Raspberry Pi je rovněž dostupný oficiální kamerový modul připojitelný přes CSI. [Raspberry Pi, 2015a]

3.1 Oficiální Raspberry Pi displej

Tento dotykový displej vytvořený speciálně pro Raspberry Pi je kompatibilní s verzemi A+, B+ a B 2. Aktuální verze Raspbianu v sobě obsahuje ovladače pro toto zařízení, zatímco Windows IoT nikoliv. Úhlopříčka tohoto displeje je 7 palců, což je srovnatelné s většinou tabletů. Rozlišení displeje je 800 x 480 pixelů. Displej využívá kapacitní technologii s desetibodovým dotykem. Barevná hloubka je 24bitová, což odpovídá 16 777 216 barvám, přičemž se tento standart někdy označuje jako True Color. Obrazovku je možné sledovat až z úhlu 70° aniž by došlo k nečitelnosti obrazu.[Raspberry Pi, 2015e]

Samotný displej je potřeba propojit s řídicí deskou, která je součástí balení. To se provádí pomocí dvou plochých kabelů. Jeden přenáší obraz, zatímco druhý přenáší signál z dotykového snímače. Řídicí desku je možné připevnit k displeji pomocí přiložených šroubků.



Obr. 3.1: Spojení řídicí desky s displejem.

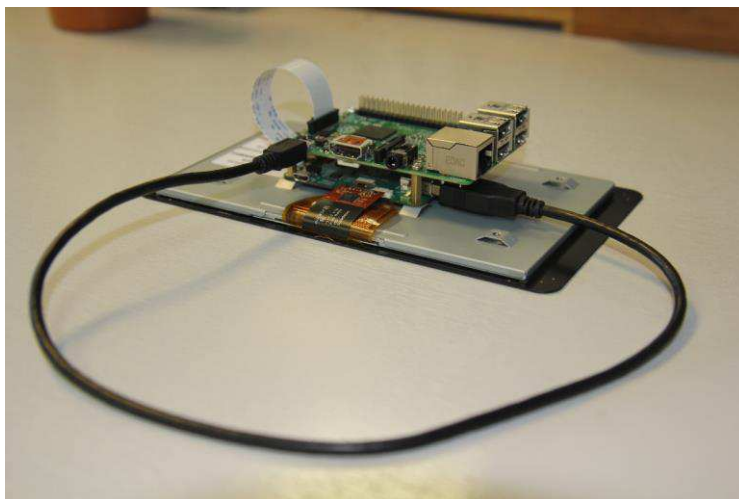
K propojení řídicí desky s Raspberry Pi slouží plochý kabel, který je připojen k DSI portu. Je třeba dávat pozor na to, že na Raspberry Pi vypadají DSI a CSI porty stejně. Správný port poznáme podle toho, že je nad ním napsáno DISPLAY, zatímco nad CSI je napsáno CAMERA. Následně můžeme Raspberry Pi přišroubovat k řídicí desce. Nakonec je potřeba zvolit způsob napájení displeje, přičemž máme na výběr ze tří možností. [Raspberry Pi, 2015e]

1. Nejjednodušší možností je připojení samostatného napájení pro displej. Výhodou tohoto zapojení je, že nám stačí zdroj, který je schopný dodat minimálně 500 mA, což splňuje většina nabíječek pro chytré telefony.



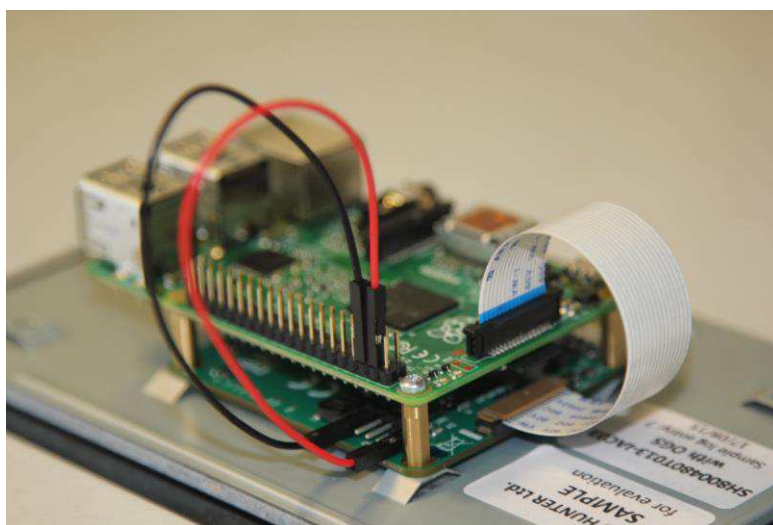
Obr. 3.2: Napájení displeje a Raspberry Pi samostatnými kabely.

2. Další možností je napájení displeje z Raspberry Pi pomocí USB. Toto zapojení je výhodné, pokud máme k dispozici kabel s jednou koncovkou USB a druhou mikro USB. Nevýhodou je, že potřebujeme zdroj, který je schopný dodat alespoň 2 A. Ten je potřeba zapojit na řídicí desce do konektoru označeného PWR IN a následně propojit konektor PWR OUT na řídicí desce s konektorem PWR IN na Raspberry Pi.



Obr. 3.3: Napájení pomocí USB spojení. [Raspberry Pi, 2015e]

3. Poslední možností je napájení pomocí GPIO portu. V tomto případě musíme napájet Raspberry Pi zdrojem s výstupem 2 A. Následně musíme propojit pin 2 na GPIO portu s pinem 5V na řídicí desce a pin 6 GPIO portu s pinem GND řídicí desky.



Obr. 3.4: Napájení z GPIO portu. [Raspberry Pi, 2015e]

3.2 Připojení k WiFi

Pro připojení k bezdrátové síti slouží USB adaptéry, někdy označovány jako tzv. dongly. Pokud používáme Raspbian, můžeme použít model od kteréhokoliv výrobce, nicméně s Windows IoT pracuje zatím pouze oficiální adaptér od Raspberry. Tato zařízení většinou pracují se standardem 802.11b/g/n s maximální přenosovou rychlostí 150 Mb/s. Při rozměrech pouze 30x16x8 mm a dosahu signálu několik desítek metrů se jedná o ideální zařízení pro technologii IoT. [Raspberry Pi, 2015f]



Obr. 3.5: Oficiální Raspberry Pi WiFi adaptér.[Swag store, 2015]

Tato zařízení stačí jednoduše zapojit do USB portu. Pokud máme k dispozici uživatelské rozhraní operačního systému, zobrazí se seznam dostupných připojení. Klikneme na síť, ke které se chceme připojit a pokud je to vyžadováno, zadáme uživatelské jméno a heslo.

Pokud máme k dispozici pouze příkazový řádek, zjistíme nejprve dostupné sítě zadáním příkazu `sudo iwlist wlan0 scan`. Najdeme v seznamu pod položkou SSID název naší žádané sítě. Poté si otevřeme v textovém editoru Nano konfigurační soubor `wpa-supPLICANT` příkazem `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`. Do něj přidáme na konec následující příkazy:

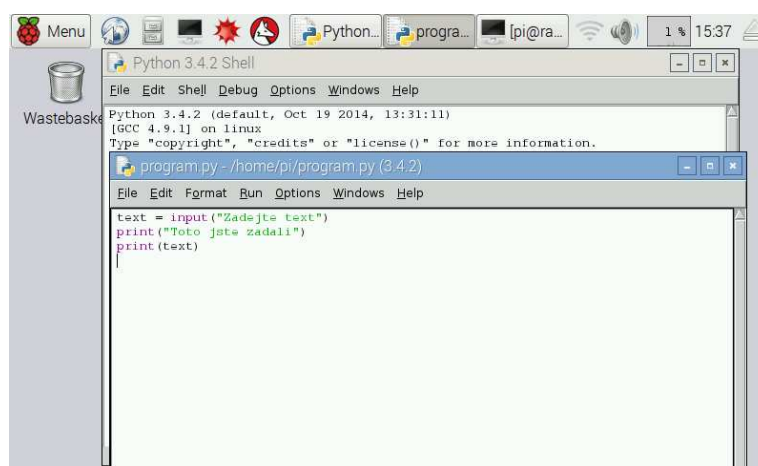
```
network={  
    ssid="Název sítě"  
    psk="Heslo"  
}
```

Soubor uložíme, a pokud nedojde k automatickému připojení, provedeme restart příkazem `sudo reboot`. [Raspberry Pi, 2015g]

4 Vývojářské možnosti

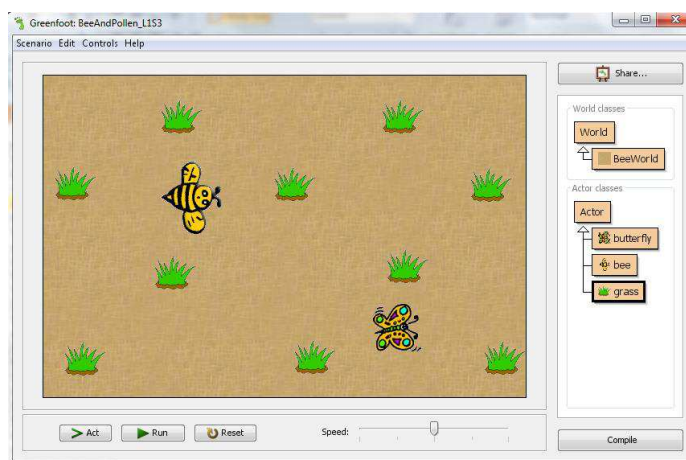
Nejjednodušší vývoj aplikací pro Raspberry Pi lze realizovat přímo na Raspberry Pi. Stačí jako operační systém použít Raspbian, který má předinstalovaných několik vývojových prostředí.

Pro vývoj v Pythonu je k dispozici vývojové prostředí IDLE 3. Jedná se o jednoduché prostředí vhodné pro začátečníky nebo výuku. Pro zjednodušení práce je vybaveno zvýrazňováním syntaxí, automatickým doplňováním a chytrým odsazováním. Ke kontrole kódu slouží integrovaný debugger s pomocnými funkcemi, jako například krokování. [Python documentation, 2015]



Obr. 4.1: Vývojové prostředí IDLE 3.

Pro vývoj v Javě je dostupné vývojové prostředí BlueJ, které nabízí všechny běžné funkce IDE. Pro Javu je dostupné i vývojové prostředí Greenfoot, které umožňuje vytvořit prostor, přidat do něj objekty a nastavit jejich pohyb. [Raspberry Pi, 2015h]



Obr. 4.2: Vývojové prostředí Greenfoot. [Oracle academy, 2015]

Pro vývoj velmi jednoduchých aplikací a animací bez znalosti programovacího jazyka lze použít Scratch. Jedná se o vizuální nástroj, který funguje na principu drag-and-drop. Skládáním jednotlivých bloků za sebe se vytváří struktura kódu. [Raspberry Pi, 2015ch]



Obr. 4.3: Vývojové prostředí Scratch. [Education Scotland, 2015]

4.1 Kivy framework

Kivy je framework pro Python, který slouží pro vývoj dotykových aplikací. Pro jeho instalaci máme na výběr ze dvou možností. Buď můžeme stáhnout obraz KivyPie, což je Linuxová distribuce založená na Rasbianu, a nahrát ho na microSD kartu. Pak stačí pouze zapnout Raspberry Pi a můžeme začít programovat. Druhou možností je nainstalovat Kivy přímo do Raspbianu. Nejprve je potřeba nainstalovat soubory, na kterých je Kivy závislé. To provedeme zadáním následujících příkazů do terminálu:

```
sudo apt-get update
```

```
sudo apt-get install libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev libsdl2-ttf-dev \
    pkg-config libgl1-mesa-dev libgles2-mesa-dev \
    python-setuptools libgststreamer1.0-dev git-core \
    gstreamer1.0-plugins-{bad,base,good,ugly} \
    gstreamer1.0-{omx,alsa} python-dev cython \
```

Následuje globální instalace samotného Kivy:

```
sudo pip install git+https://github.com/kivy/kivy.git@master
```

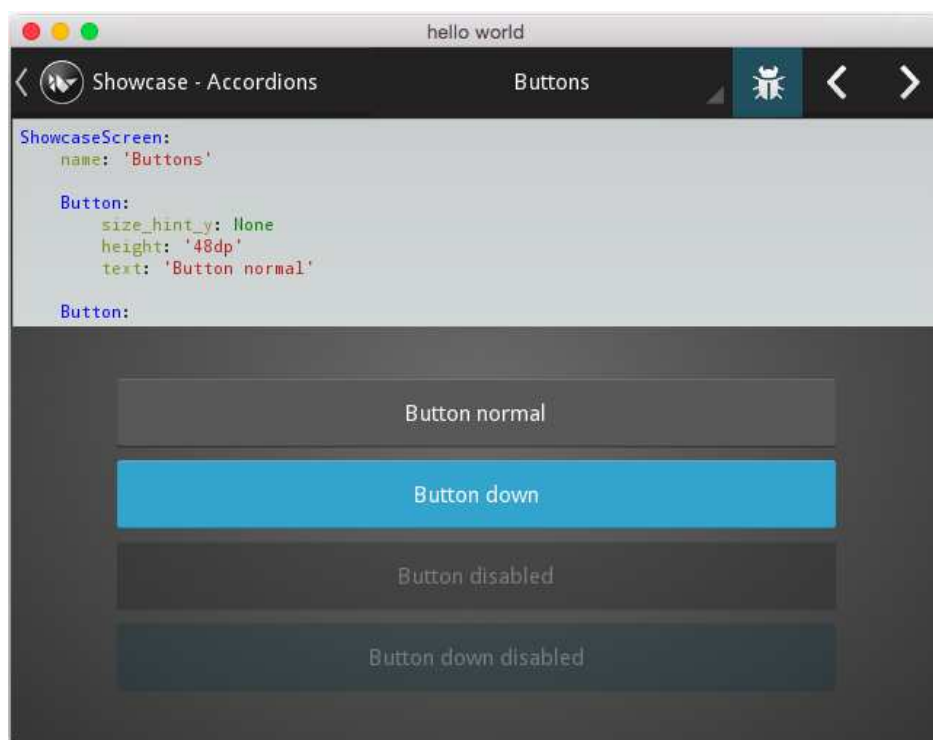
Dále je potřeba nakonfigurovat Kivy tak, aby používalo dotykový LCD displej jako vstup. To provedeme editací souboru `~/.kivy/config.ini`. Ve kterém přidáme do sekce `[input]` následující řádky:

```
mouse = mouse
```

```
mtdev_%(name)s = probesysfs,provider=mtdev
```

```
hid_%(name)s = probesysfs,provider=hidinput
```

Jamkmile máme Kivy nainstalované, můžeme si spustit ukázky, které se nacházejí v adresáři `kivy/examples`. Pro showcase je potřeba dodatečně nainstalovat další nástroje příkazem `sudo pip install pygments docutils`. [Kivy documentation, 2015a]



Obr. 4.4: Ukázka tlačítek Kivy. [Kivy documentation, 2015b]

Pokud chceme programovat vlastní aplikaci, vytvoříme soubor, který nazveme třeba `main.py`. V kódu je nejprve potřeba určit verzi Kivy, pro kterou je aplikace psaná:

```
#:kivy 1.9.2
```

Dále následuje import tříd ze souborů Kivy:

```
from kivy.app import App
```

```
from kivy.uix.label import Label
```

Soubory `uix` obsahují prvky grafického uživatelského rozhraní. Další krok je definování základní třídy naší aplikace:

```
class MyApp(App):
```

Zde si inicializujeme a vrátíme žádané widgety:

```
def build(self):
```

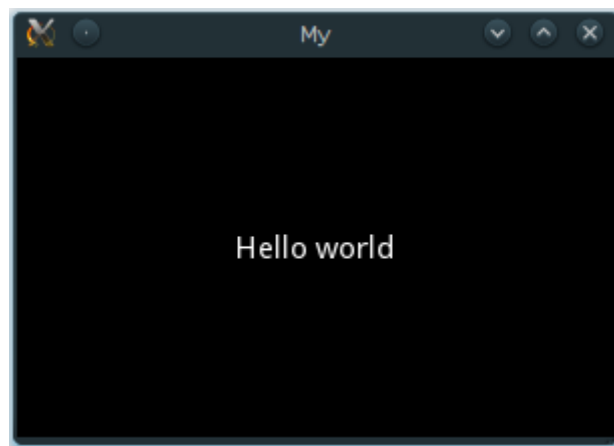
```
    return label(text='Hello world')
```

Nakonec přidáme řádky pro spuštění aplikace

```
if __name__ == '__main__':
```

```
    MyApp().run()
```

Soubor pak stačí uložit a lze jej spustit jako kterýkoliv jiný program, třeba v terminálu pomocí příkazu *python main.py*. [Kivy documentation, 2015c]



Obr. 4.5: Výsledná aplikace.[Kivy documentation, 2015c]

4.2 Mono framework

Mono je open source implementací .NET frameworku. Umožňuje kompilovat a spouštět aplikace napsané v C#. V současné době podporuje Mono .NET verze 4.5. Instalaci provedeme příkazem:

```
sudo apt-get install mono-complete
```

V Monu lze vytvořit i aplikace s podporou grafického uživatelského rozhraní. K tomu slouží sada nástrojů Gtk#, kterou nainstalujeme příkazem:

```
Sudo apt-get install gtk-sharp2
```

Pomocí textového editoru Nano vytvoříme nový soubor, který pojmenujeme např. aplikace.cs. Do něj vložíme následující kód:

```
using System;
```

```
using Gtk;
```

```

public class Hello_world
{
    static public void Main ()
    {
        Application Init ();

        Window okno = new Window("Okno");
        okno.Resize(400,200);

        Label mujtext = new Label();
        mujtext.Text = "Hello World";

        okno.Add(mujtext);
        okno.ShowAll();

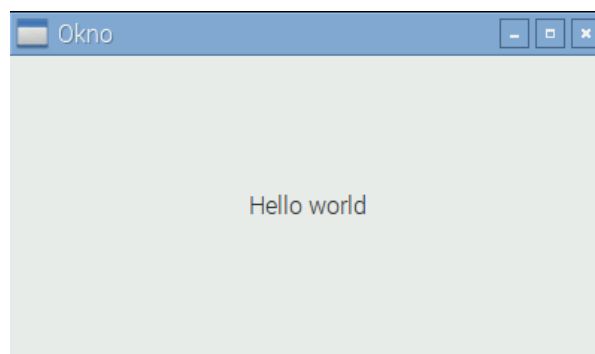
        Application.Run()
    }
}

```

Tímto jsme vytvořili třídu Hello_world, která ve funkci Main vytvoří nové okno, kterému nastaví velikost 400x200 pixelů, a v něm textové pole s textem Hello world. Program zkompilejeme příkazem:

```
mcs -pkg:gtk-sharp-2.0 aplikace.cs
```

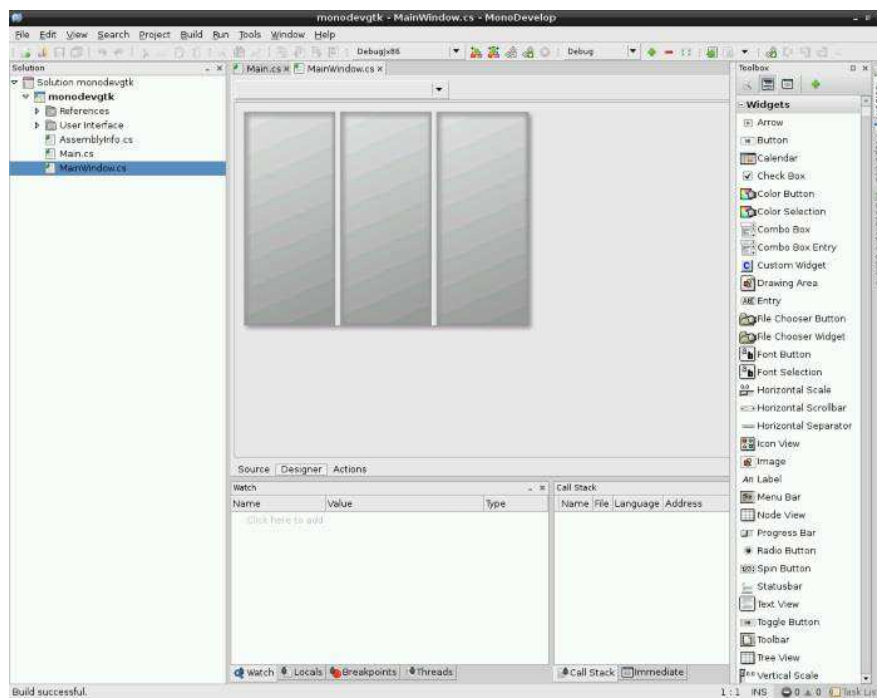
Tím se vytvoří soubor aplikace.exe. Pak už jen stačí program spustit. [Sims, 2014] [Mono, 2015]



Obr. 4.6: Výsledná aplikace.

Pro Mono je dostupné vývojové prostředí MonoDevelop, které obsahuje všechny standardní nástroje IDE. Lze jej nainstalovat příkazem:

sudo apt-get install monodevelop



Obr. 4.7: Prostředí programu MonoDevelop. [Sims, 2014]

5 Návrh modelu pro demonstrační aplikaci

V této kapitole je proveden návrh modelu domu pro demonstrační aplikaci. Cílem aplikace je ukázat možnosti Raspberry Pi 2 na příkladu automatizované domácnosti. Model je vytvořen ze stavebnice LEGO tak, že ho lze otevřít a nahlédnout dovnitř. Dům je jednopatrový a je rozdělen do čtyř místností. Dovnitř jsou přidány elektronické komponenty, které jsou řízeny pomocí Raspberry Pi 2 přes GPIO port.



Obr. 5.1: Model domu.

5.1 Obsluha dveří a vrat

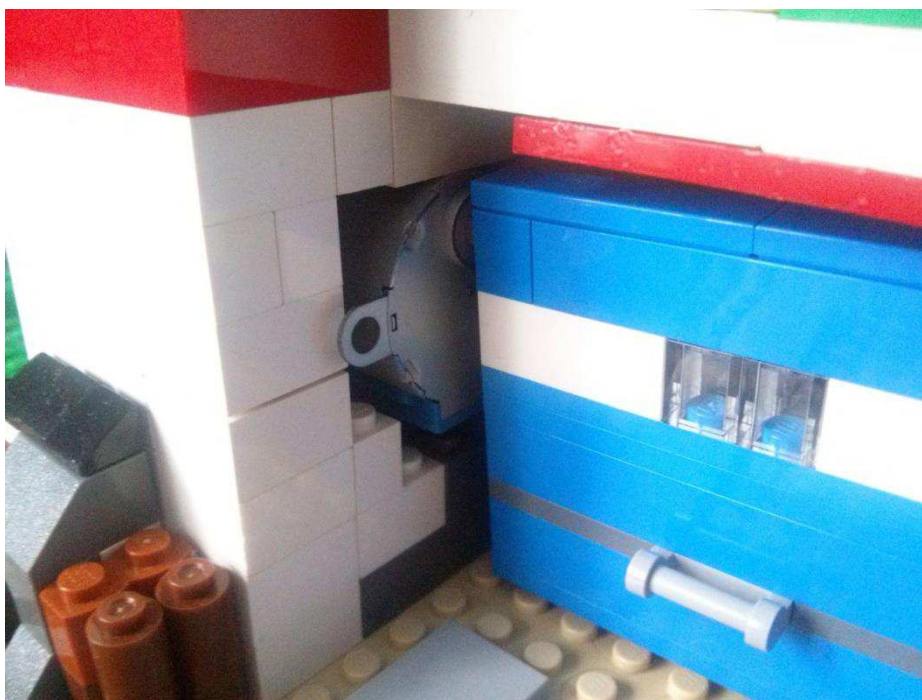
Ovládání dveří a vrat garáže slouží k demonstraci ovládání krokových motorů. Ve všech třech případech jsem použil hybridní krokový motor 28BYJ-48-5V. Pro motory jsem použil řídicí desku s Darlingtonovým polem ULN2003APC, která je navíc osazena čtyřmi LED diodami, které se rozsvěcují podle napájených fází.

Vchodové dveře jsou řízeny čtyřtakově s napájením vždy pouze jedné fáze, což snižuje energetickou náročnost a pro takto malé zatížení je dostatečné. Přenos pohybu je realizován pomocí dvou ramen spojených kloubem. Pro zjednodušení konstrukce je druhé rameno tvořeno drátem, který je omotán kolem dveří a kolem prvního ramene, tvořeného LEGO kostkou ve tvaru L.



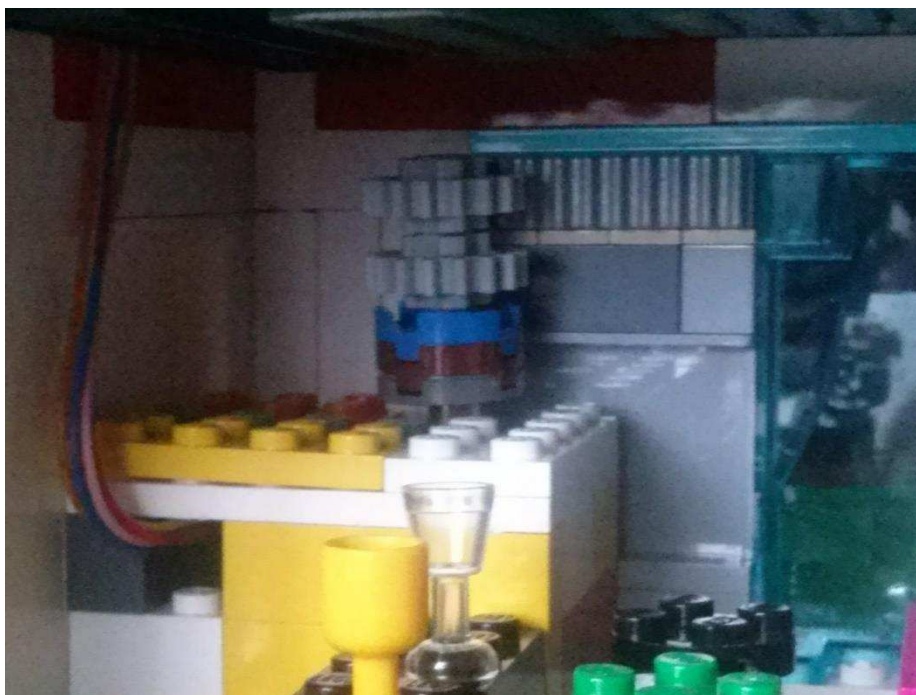
Obr. 5.2: Detail mechanismu vchodových dveří.

Garážová vrata jsou napájena také čtyřtakově, ale s napájením dvou fází, čímž se docílí většího momentu motoru za cenu větší spotřeby energie. Motor je usazen ve zdi domu a jeho hřídel je přímo napojena na osu vrat. Tento způsob je sice náročnější na usazení, nicméně odpadá nutnost převodového mechanismu.



Obr. 5.3: Garážová vrata.

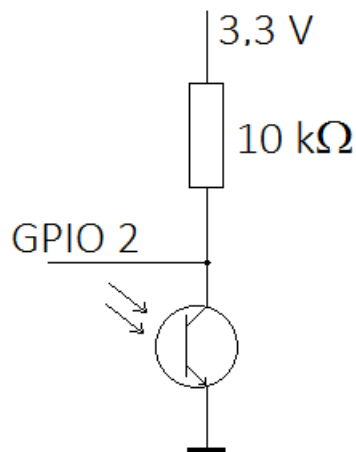
Pro otevírání posuvných dveří do zahrady jsem navrhnul mechanismus, který pro transformaci pohybu z rotačního na translační využívá kombinace pastorku a ozubeného hřebene. Výhodou tohoto zapojení je jednoduchost a relativně malá zátěž pro motorek. Nevýhodou je nutnost většího počtu kroků motoru a s tím související delší doba otevírání nebo zavírání dveří.



Obr. 5.4: Ozubení dveří do zahrady.

5.2 Obsluha osvětlení

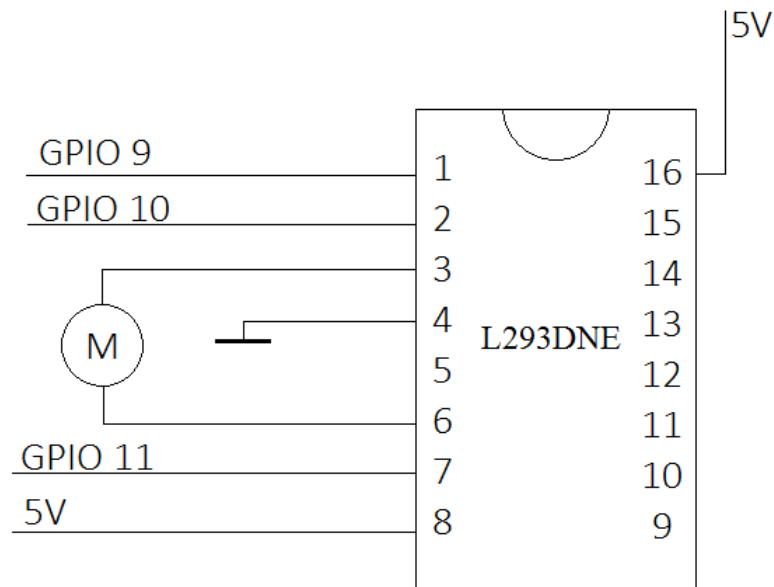
Pro osvětlení modelu jsem vybral 3mm čiré LED diody. Každá ze čtyř místností obsahuje jednu LED diodu. Jednotlivé diody lze zapínat a vypínat a u každé diody je možné nastavit automatické zapínání. V případě automatického zapínání kontroluje Raspberry Pi logickou hodnotu vstupu s fototranzistorem. Pokud Raspberry Pi dostane z fototranzistoru signál logické 1, zapne světla. Pro fototranzistor jsme použil zapojení s pull-up rezistorem o velikosti 10 k Ω . Vzhledem k tomu, že fototranzistor reaguje pouze na infračervené záření, je běžné osvětlení viditelné pro lidské oko nedostatečné. Pro správnou funkci jako čidlo denního světla je potřeba přímé sluneční záření nebo lampa s plným spektrem světla.



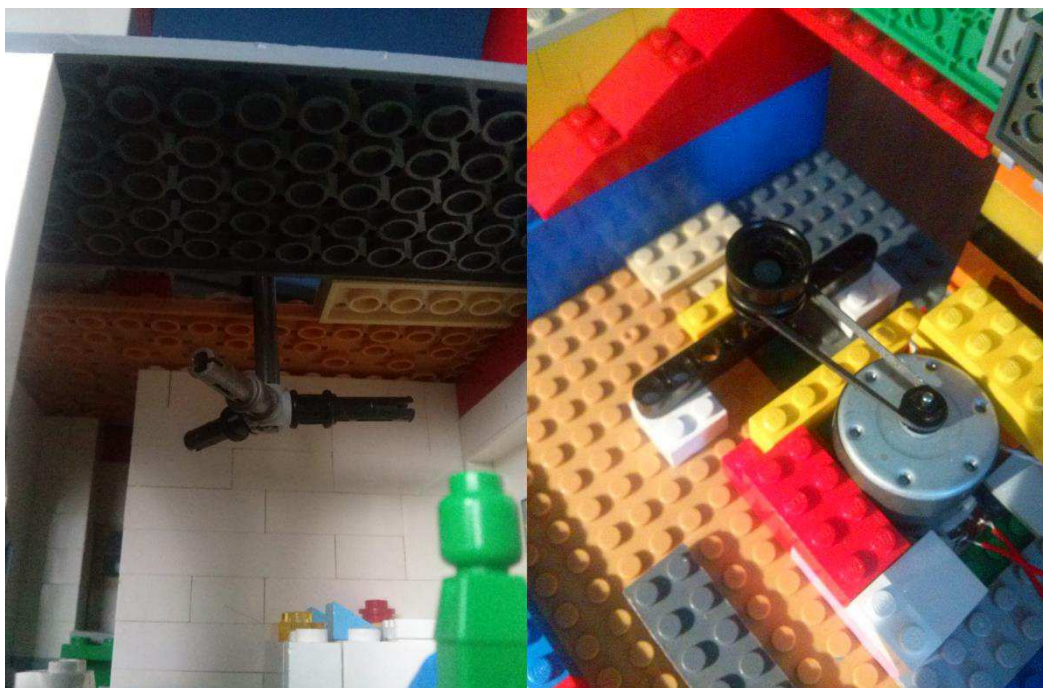
Obr. 5.5: Zapojení fototranzistoru.

5.3 Obsluha ventilátoru

Ventilátor slouží k demonstraci řízení stejnosměrného motoru. Použil jsem motor SANKO M9S60U24-1. Vzhledem k tomu, že nebylo možné napojit motor přímo na LEGO použil jsem k přenosu otáček řemenový převod z gumičky. Motor je připojen k Raspberry Pi 2 přes H-most L293DNE. Rychlost otáčení je řízena pomocí PWM o frekvenci 100 Hz. Pomocí dvou řídicích signálů lze ovládat směr otáčení motoru.



Obr. 5.6: Zapojení DC motoru.



Obr. 5.7: Umístění DC motoru.

6 Návrh grafického rozhraní demonstrační aplikace

Cílem této kapitoly je popis grafického rozhraní vytvořené řídicí aplikace pro model domu, na které jsou ukázány možnosti Raspberry Pi 2 s dotykovým LCD displejem. Pro ovládání je vytvořeno grafické rozhraní v jazyce Kv, které je uloženo v souboru `app.kv`, jenž si hlavní aplikace automaticky vyhledá. Uživatelské rozhraní je rozděleno na dvě části. První je obrazovka pro zadání PINu a druhá je obrazovka samotného ovládání. Toto je řešeno pomocí widgetu *Rozhrani*, který dědí z widgetu *ScreenManager* v hlavní aplikaci `app.py`.

<*Rozhrani*>:

Login_obrazovka:

Hlavni_obrazovka:

6.1 Úvodní obrazovka

Po spuštění aplikace se uživateli zobrazí obrazovka pro zadání PIN kódu. Pro účely vývoje je PIN nastaven na 1234. Po zadání správného kódu je uživatel vpuštěn do systému. Úvodní obrazovka slouží jako příklad tzv. nested layoutu, tedy layoutu vloženého dovnitř jiného layoutu. Pro úvodní obrazovku je použit vertikální *BoxLayout*, který osahuje 3 prvky. Prvním z nich je *Label* s textem vyzývající uživatele k zadání PIN kódu. Druhý prvek je textové pole, do kterého se zapisuje PIN a který má parametr *password = True*, což způsobuje, že čísla vepsaná do textboxu se zobrazují jako hvězdičky. Pro textové pole bylo potřeba nastavit parametr *id*, který je použit pro identifikaci, aby mohl být jeho parametr *text* upravován tlačítky.

TextInput:

password: True

id: heslobox

Třetím prvkem je *GridLayout* tvořený třemi sloupci, který v sobě obsahuje 10 tlačítek pro čísla, tlačítko pro reset textového pole a tlačítko pro potvrzení. Tlačítka číselníku fungují na principu přidávání znaků do stringu textového pole v parametru *on_press*.

on_press: heslobox.text += "0"

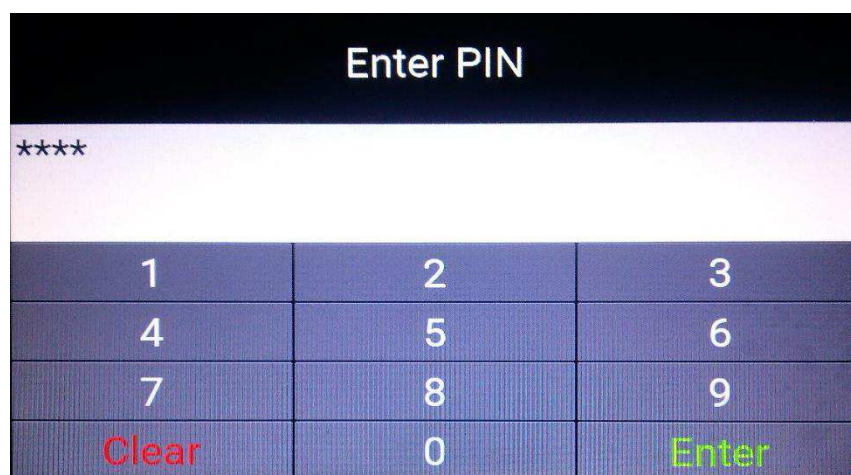
Stisk tlačítka Clear má za následek nahrazení parametru *text* textového pole prázdným stringem, čímž může uživatel vyresetovat špatně zadané heslo.

```
on_press: heslobox.text = ""
```

Při stisku tlačítka Enter dojde k vyhodnocení parametru *text* textového pole s PINem. Pokud odpovídá nastavené hodnotě, v tomto případě hodnotě 1234, dojde pomocí *ScreenManageru* ke změně obrazovky z *login* na *hlavni*. V případě, že uživatel zadá chybný kód, dojde k jeho vynulování a uživatel má možnost zadat jej znovu. Toto chování je realizováno pomocí jednoduché if-else podmínky v parametru *on_release* tlačítka.

```
on_release:
```

```
if heslobox.text == "1234": app.root.current = 'hlavni'
else: heslobox.text = ""
```



Obr. 6.1: Úvodní obrazovka.

6.2 Hlavní obrazovka

Hlavní obrazovka je tvořena widgetem *Carousel*, který umožňuje přesun mezi více obrazovkami pomocí tažení prstu do stran. Ačkoli se z programového hlediska nejedná o obrazovky, protože ty se ovládají pomocí *ScreenManageru*, budu je pro zjednodušení dále v textu takto označovat. Parametr *loop = True* slouží k zacyklení obrazovek. *Carousel* obsahuje čtyři obrazovky. První umožňuje uživateli ovládat osvětlení domku. Je tvořena vertikálním *BoxLayoutem*, který obsahuje *Label* s názvem stránky a *GridLayout*

GridLayout je tvořen čtyřmi řádky a čtyřmi sloupci. Každý řádek odpovídá jedné místnosti. V prvním slupci je vždy *Label* s názvem místností, v dalších jsou tlačítka pro

ovládání osvětlení. Každému tlačítku je přiřazen text a barva. Následně jsou definovány akce při změně stavu tlačítka. Nejprve dojde ke callbacku funkce, která ovládá GPIO pin, definované v souboru `app.py`. Následně je řešeno ošetření situace, kdy uživatel stiskne tlačítko, které je již aktivní. Při přednastaveném chování by došlo ke zrušení označení tlačítka a uživatel by pak nevěděl, v jakém stavu je světlo. Problém byl vyřešen deaktivací tlačítka a reaktivací zbylých dvou tlačítek ve skupině, k čemuž dojde při stisku.

on_state:

```
app.root.obyvakon()
```

```
self.disabled = True
```

```
lr0.disabled = False
```

```
lra.disabled = False
```

group: "LR"

id: lr1

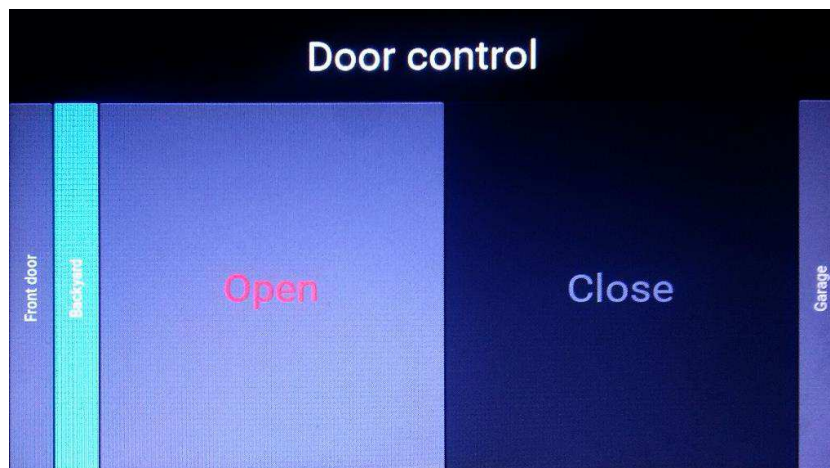
Zbýlá tlačítka jsou realizována obdobně, přičemž v jednom řádku je vždy tlačítko pro zapnutí, vypnutí a automatický režim světla.

Lights control			
Living room	On	Off	Auto
Kitchen	On	Off	Auto
Hall	On	Off	Auto
Garage	On	Off	Auto

Obr. 6.2: Ovládání osvětlení.

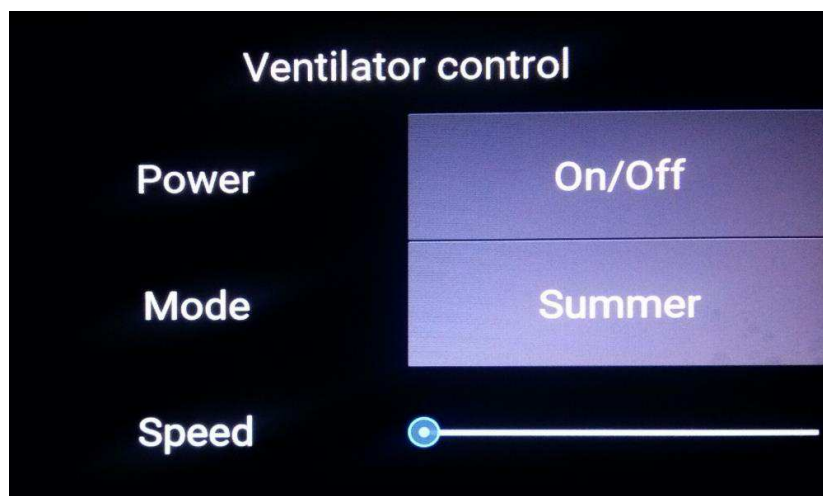
Další obrazovka má za účel ovládat dveře do domu. Vzhledem k tomu, že ovládání krokových motorů, ať už jde o rychlost otáčení nebo počet kroků, je nastaveno v souboru programové logiky `app.py`, v uživatelském rozhraní stačí vytvořit přepínač pro otevření a přepínač pro zavření dveří. Podobně jako u obsluhy osvětlení je potřeba zajistit, aby nedošlo ke zrušení označení přepínače. Stejně řešení má navíc tu výhodu,

že zabrání uživateli znovu stisknout stejné tlačítko, čímž by mohlo dojít např. k pokusu o otevření již otevřených dveří a následnému poškození modelu. Pro kompaktnost jsou vytvořeny záložky widgetu *Accordion*, přičemž každá záložka v sobě obsahuje *BoxLayout* s ovládacími prvky pro jedny dveře.



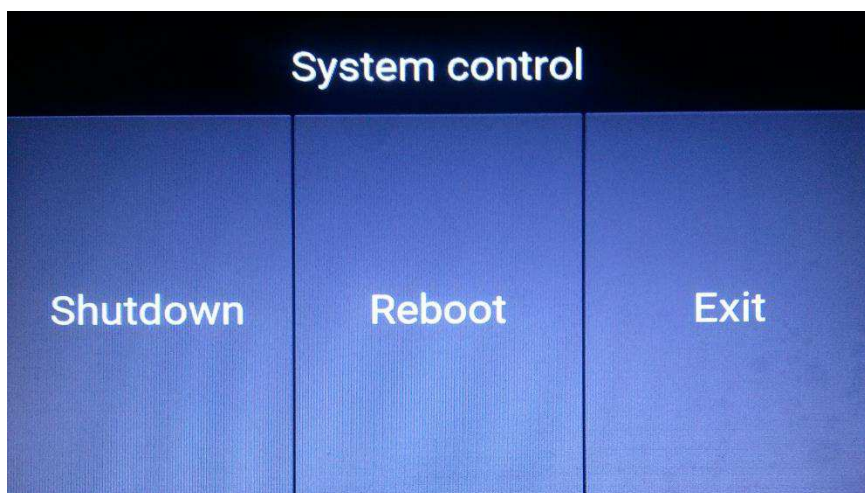
Obr. 6.3: Ovládání dveří.

Následující obrazovka slouží k ovládání ventilátoru. Je rozdělena na dva sloupce. V levém jsou *Labely* s označením prvků, které se nacházejí vpravo. V pravém sloupci je nejprve *ToggleButton*, který slouží jako hlavní spínač. Pod ním je *Spinner* ve kterém si uživatel může vybrat mezi zimním a letním režimem. V letním režimu se ventilátor otáčí ve směru hodinových ručiček, zatímco v zimním režimu se otáčí proti směru hodinových ručiček. Posledním prvkem je *Slider*, kterým je ovládána rychlost otáčení. Pokud je úplně vlevo, je střída napájení 0 a motor stojí. Posouváním doprava se střída zvyšuje až do 100 %, tedy plného výkonu motoru.



Obr. 6.4: Řízení ventilátoru.

Poslední obrazovka slouží pouze k samotnému ovládání aplikace a Raspberry Pi. Nachází se zde tři tlačítka v *BoxLayoutu*. Stiskem prvního tlačítka dojde k ukončení aplikace a vypnutí Raspberry Pi. Druhé tlačítko Raspberry pouze restartuje, přičemž aplikace nebude sama automaticky spuštěna. Při stisku třetího tlačítka dojde k ukončení aplikace.



Obr. 6.5: Obrazovka systémového ovládání.

7 Návrh programové logiky demonstrační aplikace

Cílem této kapitoly je popis programové logiky vytvořené řídicí aplikace pro LEGO model domu, na které jsou ukázány možnosti Raspberry Pi 2. Aplikace ovládá jednotlivé systémy domu pomocí binárního GPIO portu Raspberry Pi 2. Veškerá programová logika je obsažena v souboru `app.py`. Aplikace se spouští v terminálu příkazem `python app.py`.

Aplikace nejprve provede import dat operačního systému, času, ovladače GPIO a prvků ze souborů Kivy. Dále jsou přiřazeny názvy jednotlivým pinům GPIO portu viz tabulka 7.1 a dojde k jejich nastavení jako vstup nebo výstup.

Tab. 7.1: Nastavení GPIO pinů.

Číslo pinu	Název pinu	Popis
2	dioda	Vstup z fototranzistoru
4	obytak	Výstupy pro jednotlivé LED diody
17	kuchyn	
27	hala	
22	garaz	
18	motor1a	Fáze motoru vchodových dveří
23	motor1b	
24	motor1c	
25	motor1d	
5	motor2a	Fáze motoru dveří do zahrady
6	motor2b	
13	motor2c	
19	motor2d	
12	motor3a	Fáze motoru garážových vrat
16	motor3b	
20	motor3c	
21	motor3d	
9	pwmpin	Řízení rychlosti DC motoru
10	ssleto	Ovládání směru DC motoru
11	sszima	

Následuje první třída aplikace s názvem *Autosviceni*, která obsahuje jedinou funkci s názvem *sviti*. Tato funkce je volána každou polovinu sekundy a kontroluje, zda je v grafickém rozhraní u jednotlivých místností zvolena možnost automatického osvětlení a pokud ano, zkontroluje logickou hodnotu vstupu foto diody. Pokud jsou obě podmínky splněny, dojde k přepnutí stavu pinu pro osvětlení odpovídající místosti do

logické 1. Jakmile jedna z podmínek nebo obě současně platit přestanou, dojde k přepnutí do logické 0.

```
if autoobyvak == True:
```

```
    if GPIO.input(dioda) == True:
```

```
        GPIO.output(obyvak, GPIO.HIGH)
```

```
    else:
```

```
        GPIO.output(obyvak, GPIO.LOW)
```

Tab. 7.2: Pravdivostní tabulka pro automatické svícení.

Stav tlačítka	Stav fototranzistoru	Stav osvětlení
1	1	1
1	0	0
0	1	0
0	0	0

Dálší třídou je *Rozhrani*, jenž dědí z Kivy prvku pro přesun mezi obrazovkami *ScreenManager*. Tato třída obsahuje veškeré zbývající funkce pro ovládání GPIO pinů z uživatelského rozhraní. Nejprve jsou zde funkce pro manuální ovládání osvětlení. Každá tato funkce nejprve deaktivuje automatické svícení pokud bylo zapnuto, a poté nastaví příslušný pin do odpovídajícího stavu.

```
def obyvakon(self):
```

```
    global autoobyvak
```

```
    autoobyvak = False
```

```
    GPIO.output(obyvak, GPIO.HIGH)
```

Dále se zde nacházejí funkce pro ovládání krokových motorků. Ty jsou vytvořeny smyčkou `for`, která se opakuje podle nastaveného počtu kroků. Samotný obsah smyčky tvoří nastavování příslušných čtyř fází motoru v určité kombinaci podle zvoleného směru otáčení a typu řízení. Mezi jednotlivými změnami je nastavena prodleva 5 ms, která se experimentálně jevila jako optimální.

Následují tři funkce pro obsluhu ventilátoru. První funkce nazvaná *ventilator* zapíná a vypíná ventilátor, což provádí změnou střídy na nulu (vypnuto) nebo na poslední nastavenou hodnotu, která je uložena v proměnné *strida* (zapnuto).

```
def ventilator(self, obj, state):  
    if obj.state == "down":  
        ventilace = True  
        pwmpin.ChangeDutyCycle(strida)  
    else:  
        ventilace = False  
        pwmpin.ChangeDutyCycle(0)
```

Další funkce se stará o aktualizaci nastavení hodnoty proměnné *strida* podle jezdce uživatelského rozhraní a pokud je ventilátor spuštěn, nastaví i střídu na GPIO výstupu.

```
def zmena_stridy(self, obj, value):  
    strida = obj.value  
    if ventilace == True:  
        pwmpin.ChangeDutyCycle(strida)
```

Poslední funkce ventilátoru umožňuje nastavit směr otáčení motoru pomocí signálu pro H-můstek, přičemž vždy musí být aktivní pouze jeden ze signálů.

```
def zmena_smeru(self, obj, text):  
    if obj.text == "Summer":  
        GPIO.output(ssleto, GPIO.HIGH)  
        GPIO.output(sszima, GPIO.LOW)  
    else:  
        GPIO.output(sszima, GPIO.HIGH)  
        GPIO.output(ssleto, GPIO.LOW)
```

Zbývající tři funkce této třídy slouží k práci se samotnou aplikací. Využívají především příkazu *os.system*, který umožňuje vykonat stejné příkazy, které se zadávají

do terminálu. Tím se dá docílit vypnutí nebo restartu systému. Při ukončení aplikace bez vypnutí systému je navíc potřeba resetovat GPIO port.

def terminal(self):

GPIO.cleanup()

App.get_running_app().stop()

Poslední třídu tvoří hlavní smyčka aplikace. Zde se nejprve nastaví volání funkce *sviti* třídy *Autosviceni* a poté se vrátí obě předchozí třídy *Autosviceni* a *Rozhrani*. Aplikaci zakončují příkazy pro případ pádu aplikace nebo přerušení pomocí klávesnice, které resetují GPIO port.

Závěr

Cílem této práce bylo prozkoumat možnosti mikropočítače Raspberry Pi 2 a vytvořit aplikaci s dotykovým displejem, která tyto možnosti demonstruje. V první části bakalářské práce je charakterizován mikropočítač Raspberry Pi 2 po hardwarové stránce. Jeho hardwarové parametry jsou pak srovnány s předchozími verzemi, oproti kterým došlo ke zlepšení. Popsány jsou rovněž softwarové možnosti zařízení se zaměřením na kancelářské programy, aplikace pro média a výukový software předinstalovaný v operačním systému Raspbian.

Dále je zdokumentován proces instalace operačních systémů Raspbian a Windows 10 IoT. Nejprve je popsáno potřebné hardwarové a softwarové vybavení. Poté následuje popis samotné instalace. Ta se skládá z přípravy obrazu operačního systému, jeho zapsání na microSD kartu a spuštění v Raspberry Pi 2.

Další část se zabývá dostupnými periferiemi, především těmi, které lze nalézt u běžného stolního počítače nebo využívají rozhraní USB. Následuje popis oficiálního dotykového LCD displeje, postup jeho zprovoznění a tři možnosti jeho napájení z Raspberry Pi. Také je popsáno zprovoznění připojení k bezdrátové síti pomocí USB WiFi adaptéru.

Pro Raspberry Pi 2 jsou dokumentovány možnosti vývoje aplikací. Jsou popsána předinstalovaná vývojová prostředí v Rasbianu pro Python, Javu a další. Následuje popis instalace a vývoje jednoduché aplikace v Kivy, což je Framework pro Python k vytváření dotykových aplikací, a Mono, open source implementaci .NET frameworku.

Pro Raspberry Pi 2 a LCD displej byla vytvořena demonstrační aplikace, využívající GUI naprogramované s pomocí frameworku Kivy, pro ovládání automatizované domácnosti modelového domu ze stavebnice LEGO. Aplikace umožňuje zapínat a vypínat osvětlení domu z LED diod, což demonstruje práci s GPIO výstupy. Osvětlení lze také zapínat automaticky podle stavu napětí na fototranzistoru, což ukazuje možnosti GPIO jako vstupů. Dále aplikace umožňuje otevírání a zavírání dveří a vrat, čímž je předvedena práce s krokovými motory s různými typy řízení. K demonstraci řízení stejnosměrného motoru slouží v modelu ventilátor, u kterého lze ovládat zapínání a směr otáčení pomocí signálů pro H-most, a dále také rychlost otáčení pomocí PWM.

Ze zjištěných poznatků vyplývá, že Raspberry Pi 2 v kombinaci s dotykovým LCD displejem se nachází na pomezí mezi stolním počítačem a tabletem. Můžeme jej tedy využívat jako kompaktní stolní počítač za předpokladu, že nebudeme potřebovat výpočetně náročné operace jako např. 3D modelování, ale vystačíme si kancelářskými aplikacemi nebo internetem. Nebo ho naopak využijeme jako tablet s daleko větším výkonem, který zvládá i komplexní aplikace pro běžný tablet nedostupné.

Raspberry Pi je také ideální zařízení pro technologii IoT. To znamená, že je vhodný k propojení s dalšími zařízeními prostřednictvím internetu. Takto se dá využít např. v systémech pro automatizaci domácnosti, podobně jako je ukázáno v demonstrační aplikaci, ale s tím rozdílem, že Raspberry Pi s dotykovým LCD displejem slouží jako centrální ovládací panel, který pomocí bezdrátové sítě ovládá další zařízení, jako třeba chytrý termostat topení nebo spínače osvětlení.

Další využití může být palubní počítač v automobilech. Pomocí dotykového displeje je možné ovládat přehrávač hudby a videa nebo GPS navigaci. V kombinaci s videokamerou připevněnou na zadní nárazník je možné při zařazení zpátečky přenášet obraz, čímž se usnadní parkování. Vzhledem k tomu, že většina čerpacích stanic nabízí zdarma WiFi připojení a mobilní operátoři nabízí USB modem, je možné využít Raspberry k prohlížení internetu při hledání ubytování, restaurací apod. během cestování.

Literatura

BECHYNSKÝ, Štěpán. *Raspberry Pi 2 a Windows 10 IoT Core*. [online]. 11. 5. 2015 [cit. 2015-11-12]. Dostupné z: <https://www.zdrojak.cz/clanky/raspberry-pi-2-windows-10-iot-core/>

Designspark. *Introducing the Raspberry Pi B+*. [online]. 2015 [cit. 2015-11-25]. Dostupné z: <http://www.rs-online.com/designspark/electronics/eng/blog/introducing-the-raspberry-pi-b-plus>

Distrowatch. *Raspbian*. [online]. 2015 [cit. 2015-12-02]. Dostupné z: <http://distrowatch.com/table.php?distribution=raspbian>

Education Scotland. *Scratch*. [online]. 2015 [cit. 2016-01-05]. Dostupné z: <http://www.educationscotland.gov.uk/learningandteaching/approaches/ictineducation/gamesbasedlearning/gamedesign/scratch.asp>

Element14. *Raspberry Pi 2 which os is best*. [online]. 13. 4. 2014 [cit. 2015-11-26]. Dostupné z: <http://www.element14.com/community/polls/2103>

eLinux. *RPI Hardware History*. [online]. 2015 [cit. 2015-11-12]. Dostupné z: http://elinux.org/RPi_HardwareHistory

Exp-tech. *Raspberry Pi 2 Modell B*. [online]. 2015 [cit. 2015-11-25]. Dostupné z: <http://www.exp-tech.de/raspberry-pi-2-mod-b>

Kivy documentation. *Installation*. [online]. 2015a [cit. 2016-01-01]. Dostupné z: <http://kivy.org/docs/gettingstarted/installation.html>

Kivy documentation. *Showcase of Kivy features*. [online]. 2015b [cit. 2016-01-02]. Dostupné z: http://kivy.org/docs/examples/gen__demo__showcase__main__py.html

Mono. *Gtk sharp begginers guide*. [online]. 2015 [cit. 2016-01-10]. Dostupné z: <http://www.mono-project.com/docs/gui/gtksharp/beginners-guide/>

Oracle academy. *Getting started with Java using Greenfoot*. [online]. 2015 [cit. 2016-01-05]. Dostupné z: <http://www.oracle.com/webfolder/technetwork/tutorials/OracleAcademy/GreenfootSelfStudyV1/obe.html>

Python documentation. *2.5.5. IDLE*. [online]. 2015 [cit. 2015-12-31]. Dostupné z: <https://docs.python.org/3.6/library/idle.html>

Raspberry Pi. *Difference between BlueJ and Greenfoot*. [online] 2015h [cit. 2015-12-31]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?f=81&t=122655>

Raspberry Pi. *FAQ*. [online]. 2015a [cit. 2015-11-12]. Dostupné z: <https://www.raspberrypi.org/help/faqs/#introShares>

Raspberry Pi. *Installing operating system images*. [online]. 2015b [cit. 2015-11-12]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/installing-images/>

Raspberry Pi. *NOOBS*. [online]. 2015c [cit. 2015-12-02]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/noobs.md>

Raspberry Pi. *Raspberry Pi NOOBS Setup*. [online]. 2015d [cit. 2015-11-12]. Dostupné z: <https://www.raspberrypi.org/help/noobs-setup/>

Raspberry Pi. *Raspberry Pi USB WiFi dongle*. [online]. 2015f [cit. 2015-12-14]. Dostupné z: <https://www.raspberrypi.org/products/usb-wifi-dongle/>

Raspberry Pi. *Scratch*. [online]. 2015ch [cit. 2015-12-31]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/scratch/>

Raspberry Pi. *Setting WiFi up via the command line*. [online]. 2015g [cit. 2015-12-14]. Dostupné z: <https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>

Raspberry Pi. *The eagerly awaited Raspberry Pi display*. [online]. 2015e [cit. 2015-12-14]. Dostupné z: <https://www.raspberrypi.org/blog/the-eagerly-awaited-raspberry-pi-display/>

SIMS, Gary. *How to write C# programs on a Raspberry Pi using Mono*. [online]. 16. 6. 2014 [cit. 2016-1-10]. Dostupné z: <https://www.maketecheasier.com/write-c-sharp-programs-raspberry-pi/>

SparkFun Electronics. *Raspberry Pi – Model B*. [online]. 2015 [cit. 2015-11-12]. Dostupné z: <https://www.sparkfun.com/products/retired/11546>

Swag store. *Official Raspberry Pi WiFi dongle*. [online]. 2015 [cit. 2015-12-14]. Dostupné z: <http://swag.raspberrypi.org/products/official-raspberry-pi-wifi-dongle>

Windows dev center. *Using PowerShell to connect and configure a device running Windows 10 IoT core*. [online]. 2015a. [cit. 2015-11-30]. Dostupné z: <https://ms-iot.github.io/content/en-US/win10/samples/PowerShell.htm>

Windows dev center. *Windows IoT – Setup your Raspberry Pi 2*. [online]. 2015b. [cit. 2015-11-12]. Dostupné z: <https://ms-iot.github.io/content/en-US/win10/SetupRPI.htm>

Seznam příloh

A Diplom ze soutěže STOČ na VŠB-TUO

Obsah přiloženého CD

- Složka *Bakalářská práce a přílohy*
 - Soubor *Bakalářská práce.pdf*
 - Soubor *Příloha A – Diplom ze soutěže STOČ na VŠB-TUO.pfd*
- Složka *Aplikace*
 - Soubor *app.py*
 - Soubor *app.kv*
- Složka *Videa*
 - Soubor *Celý dům.mp4*
 - Soubor *Garáž.mp4*
 - Soubor *Přední dveře zepředu.mp4*
 - Soubor *Přední dveře zezadu.mp4*
 - Soubor *Světla – automaticky řízená.mp4*
 - Soubor *Světla – manuálně řízená.mp4*
 - Soubor *Ventilátor zespodu.mp4*
 - Soubor *Ventilátor shora.mp4*
 - Soubor *Zadní dveře zepředu.mp4*
 - Soubor *Zadní dveře zezadu.mp4*